

De Evolutionaire Aanpak Garandeert Project Succes

Niels Malotaux

N R Malotaux
Consultancy

030-228 88 68

niels@malotaux.nl

www.malotaux.nl/nrm

Wanneer is een project succesvol?

Waarvoor zijn projecten niet succesvol?

Het probleem

- **Veel projecten zijn niet direct succesvol**
- **Veel projecten zijn te laat klaar**

of, positiever:

- **Kan ik mijn project nog succesvoller afronden**
- **In kortere tijd**

Wat kunnen we daaraan doen?

- De oorzaken aanpakken
- De mens accepteren zoals hij is

Discipline

- **Beheersen van verkeerde neigingen**
 - **Al weten we zelf wat de beste manier is ...**
(als niemand het ziet ...)
 - **Discipline is gewoon moeilijk**
 - **Romeinen 7:19**
 - Ik doe niet het goede wat ik wil doen ...
- **Elkaar helpen** (over de schouder meekijken)
- **Snel succes** (dus binnen twee weken en niet over een jaar)
- **Fouten maken** (en er direct van leren)

Intuïtie

- Inzicht zonder nadenken
- Intuïtie wordt gevoed door ervaring
- Het is gratis, het werkt altijd
- We kunnen het niet eens afzetten
- Soms wijst intuïtie ons de verkeerde weg
- Vaak snapt het hoofd het wel, maar het hart nog niet
- Coaching is her-richten van intuïtie

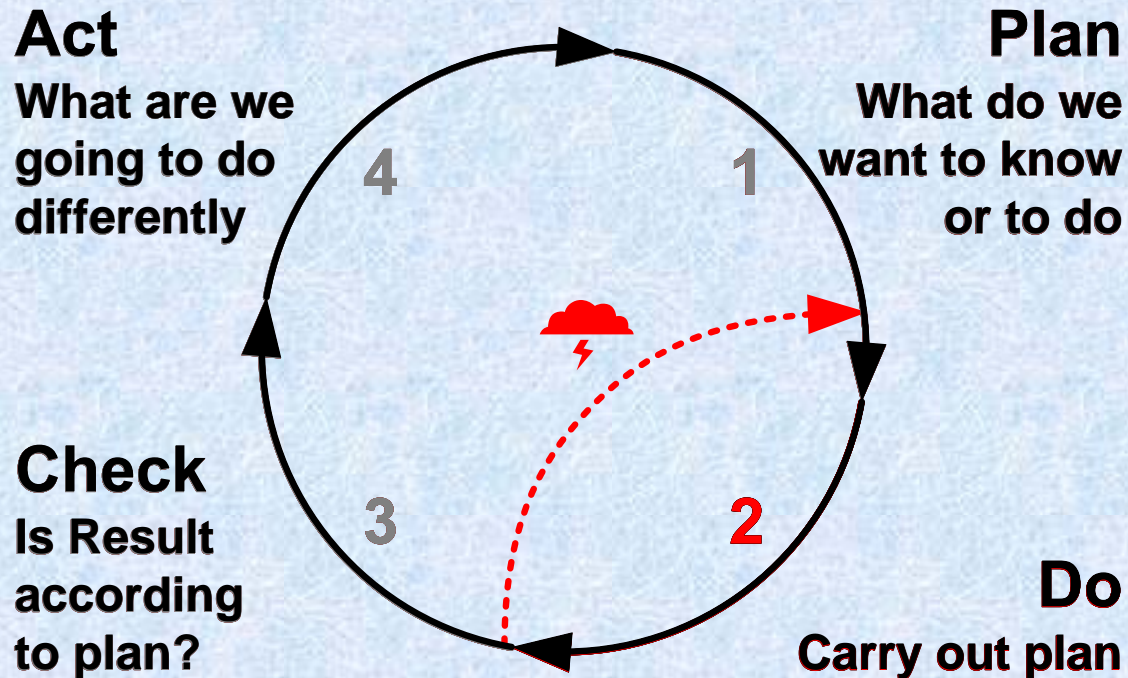
Communicatie

- Verkeersongeval: getuigen vertellen *hun* waarheid
- Zelfde woorden, andere *concepten*
- Ons brein is tevreden met fuzzy resultaten
- Leg het een collega uit
- Opschrijven is uitleggen aan papier
- Als je het opschrijft kun je het veranderen
- Mondelinge communicatie verdampt waar je bij staat
- E-mail communicatie verdampt binnen een paar dagen

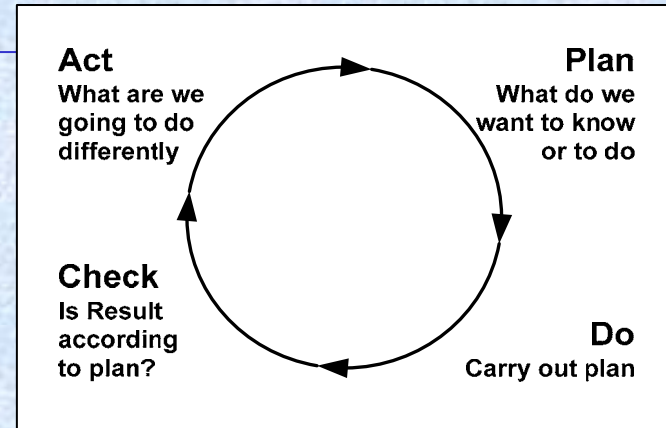
Doel van een project

- **De klant leveren**
 - wat hij nodig heeft
 - op het moment dat hij het nodig heeft
 - tot tevredenheid
 - en om succes *voller* te worden dan ervoor
- **Beperkt door**
 - wat de klant zich kan veroorloven
 - wat we tot wederzijds voordeel en tevredenheid kunnen leveren
 - in een redelijke tijd

Het geheim: de Plan-Do-Check-Act cyclus



Kennis hoe het doel te bereiken



- Met zeer korte en frequente Plan-Do-Check-Act cycli
- Steeds selecterend wat het belangrijkste is om te doen

kunnen we

- Zo snel mogelijk bepalen wat de werkelijke requirements zijn
- Zo snel mogelijk leren hoe we zo effectief en efficiënt mogelijk deze requirements implementeren

en kunnen we

- problemen eerder op het spoor komen, zodat we meer tijd hebben om er iets aan te doen

doing the
right things

doing the
right things
right

Evo

Act

What are we
going to do
differently

Plan

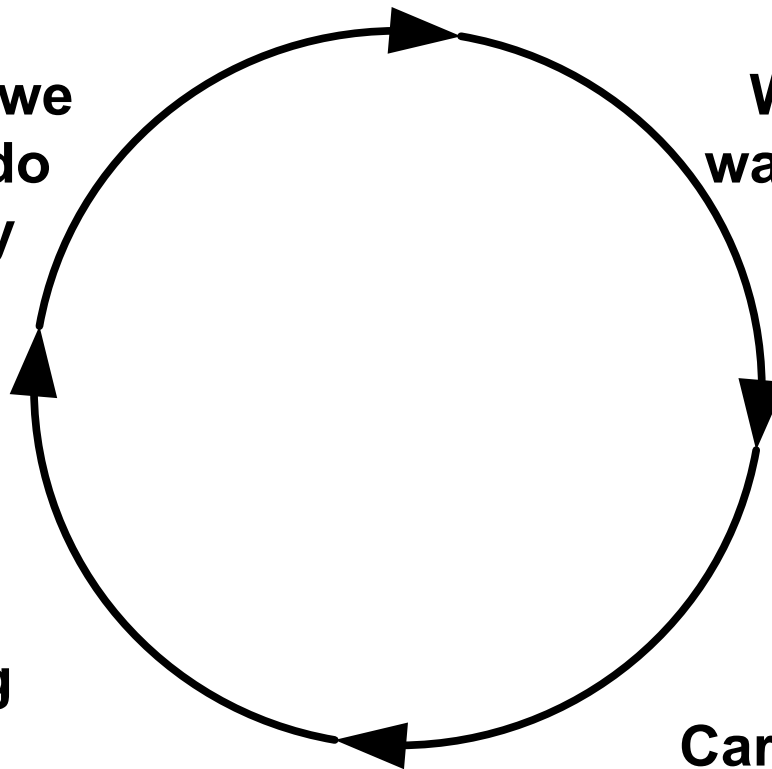
What do we
want to know
or to do

Check

Is Result
according
to plan?

Do

Carry out plan



Belangrijke Evo elementen

- **Requirements Engineering en Design**
 - Om te weten wat we voor wie moeten en gaan doen *en wat niet*
 - Het gaat uiteindelijk om **performance verbetering**
 - Functionele requirements geven aan wát we gaan verbeteren
 - Kwantificeren van requirements (Planguage)
 - Dynamisch prioriteren wat belangrijker is en wat minder
- **TaskCycle**
 - Organiseren van het werk
 - Optimaliseren van schatten
 - Nakomen van beloftes
- **DeliveryCycle**
 - Optimaliseren van requirements en verifiëren van aannames
 - Genereren van feedback om te leren
- **TimeLine**
 - Beheersen van langere tijdsperioden

Planguage

ref Tom Gilb

RQ27: Maximum Response Time

Scale: Seconds between <asking> for information and <appearance> of it.

Meter: Add a function to the software to measure the maximum response time value and the <range of values> per <working day>.

Benchmarks (Playing Field):

Past: 3 sec (our previous product)

Current: 0.6 sec [competitor y, product x, 2006] ← Marketing Survey Jan 2006

Record: 0.2 sec [competitor x, product y]

Wish: 0.2 sec [2008] ← customer's head of R&D, 19 Feb 2005, <document ...>

Note: Less than 0.2 sec is not noticed by the user,
so there is no use in trying to be better than 0.2 sec

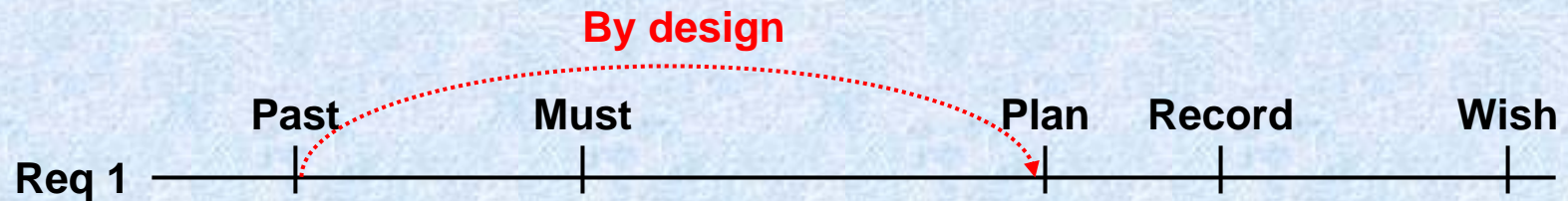
Requirements:

Must: 1 sec [99%] ← project-contract

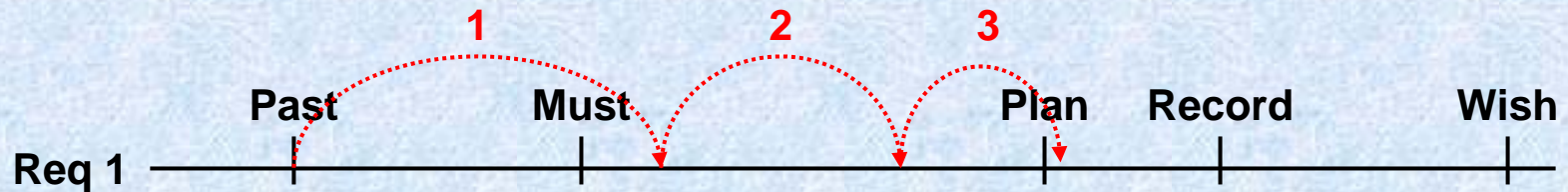
Must: 1.5 sec [100%] ← project-contract

Plan: 0.5 sec ← project-contract

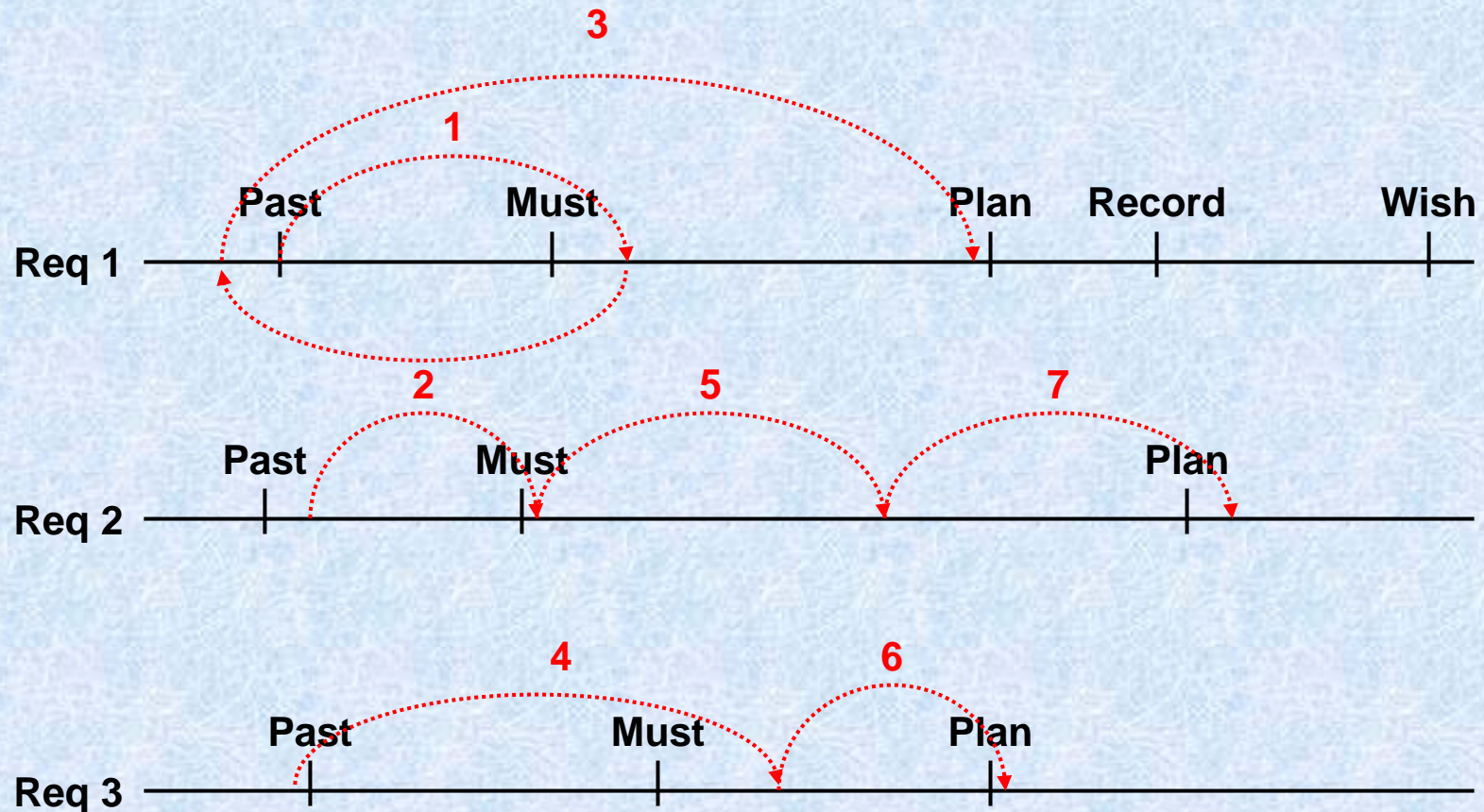
Bereiken van een doelstelling in één stap



Bereiken van een doelstelling in meer stappen

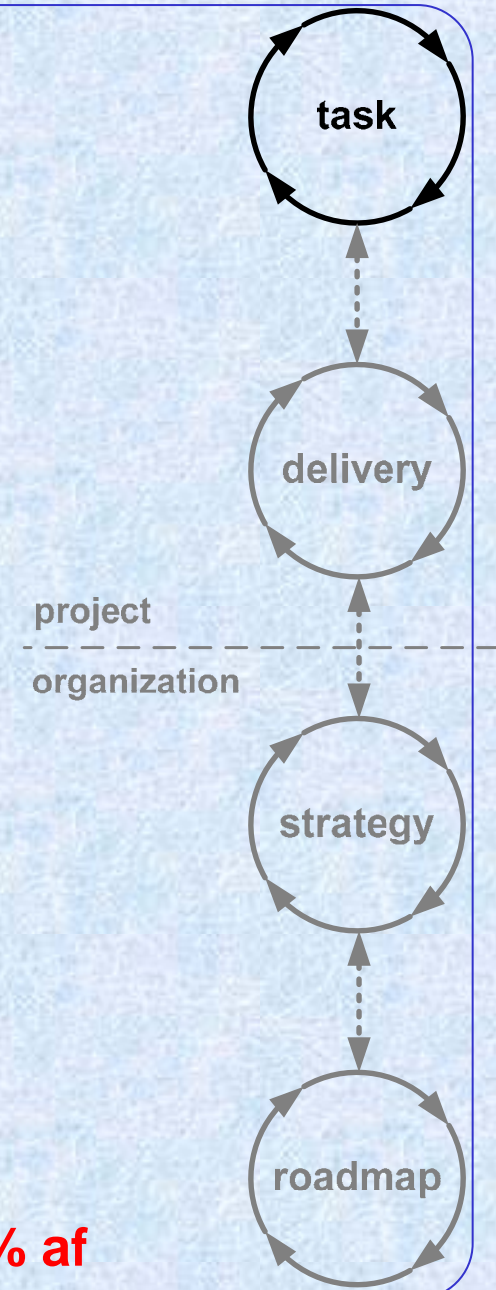


De praktijk: Bereiken van tegenstrijdige doelstellingen



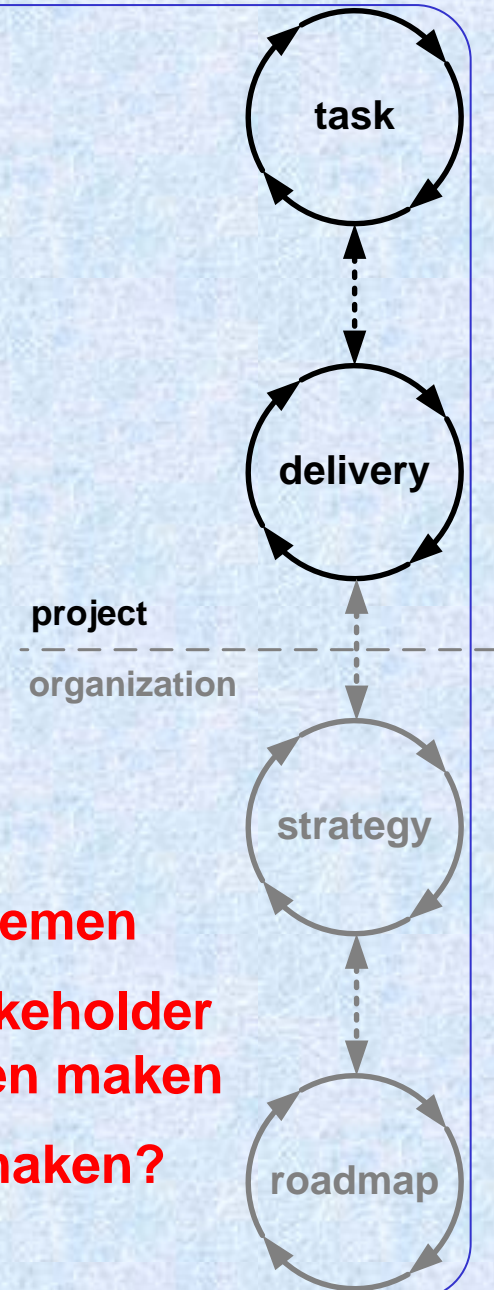
Wekelijkse TaskCycle

- *Doen we de juiste dingen, in de juiste volgorde, tot het juiste detailniveau voor dit moment*
- *Optimaliseren van onze bekwaamheid in schatten, plannen en opvolgen, zodat we de toekomst beter kunnen voorspellen*
- *We selecteren alleen de belangrijkste taken, we voeren geen minder belangrijke taken uit en werken niet aan ongedefiniëerde taken*
- *Er zijn ca 26 Planbare uren in een week (2/3)*
- *In de overige 1/3 van de tijd doen we alle andere dingen die we ook moeten doen*
- *Taken zijn aan het einde van de week altijd 100% af*

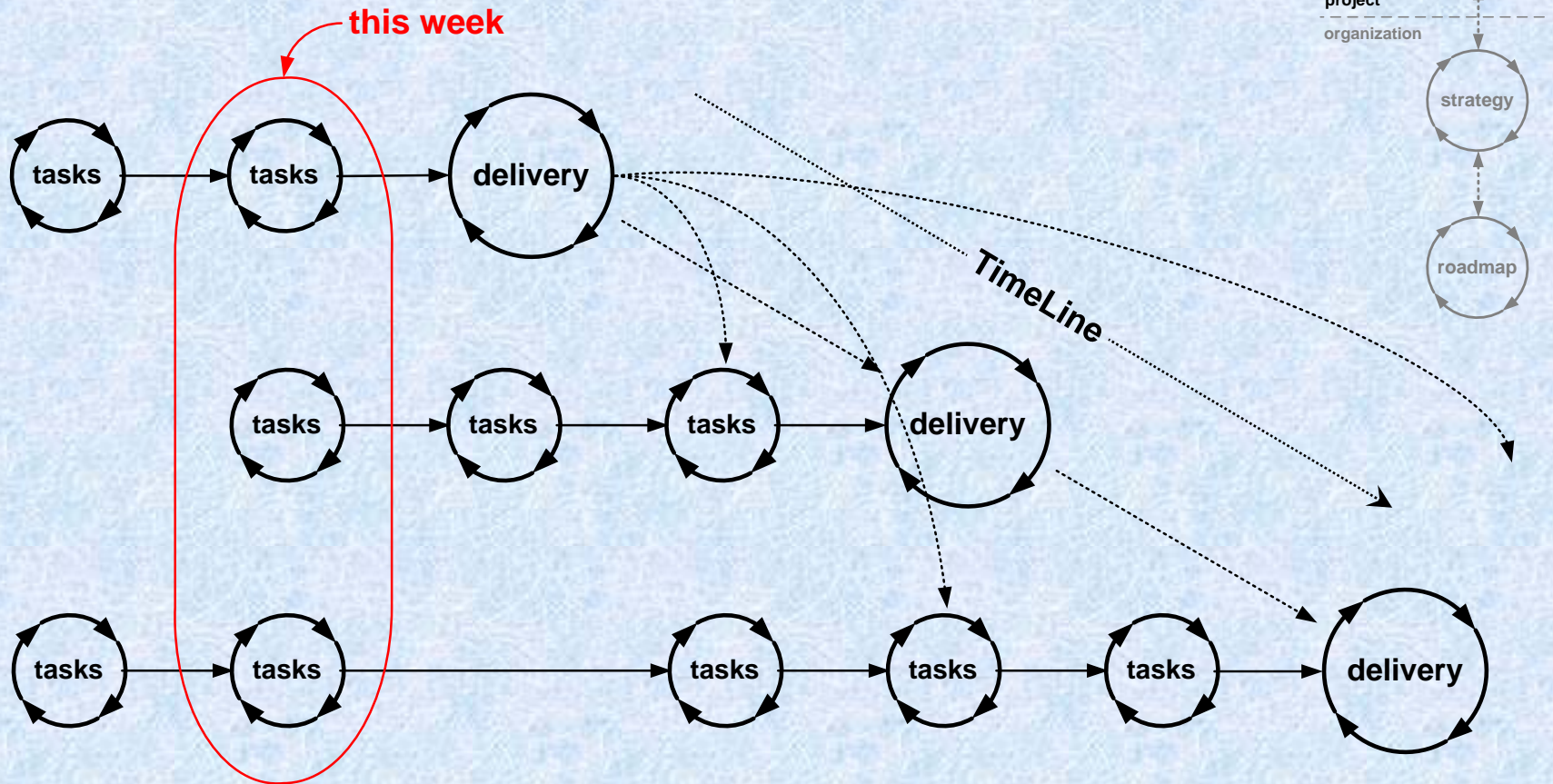


DeliveryCycle

- **Leveren we de juiste dingen, in de juiste volgorde, tot het juiste detailniveau voor dit moment**
- **Optimaliseren van de requirements en verifiëren van de aannames**
- **Optimale feedback**
- **We leveren alleen aan Stakeholders die *gretig klaar staan* om ons resultaat in ontvangst te nemen**
- **Leveren van de lekkerste en belangrijkste Stakeholder values die we met minimale investering kunnen maken**
- **Kunnen we Stakeholders nu al productiever maken?**
- **Nooit langer dan 2 weken**

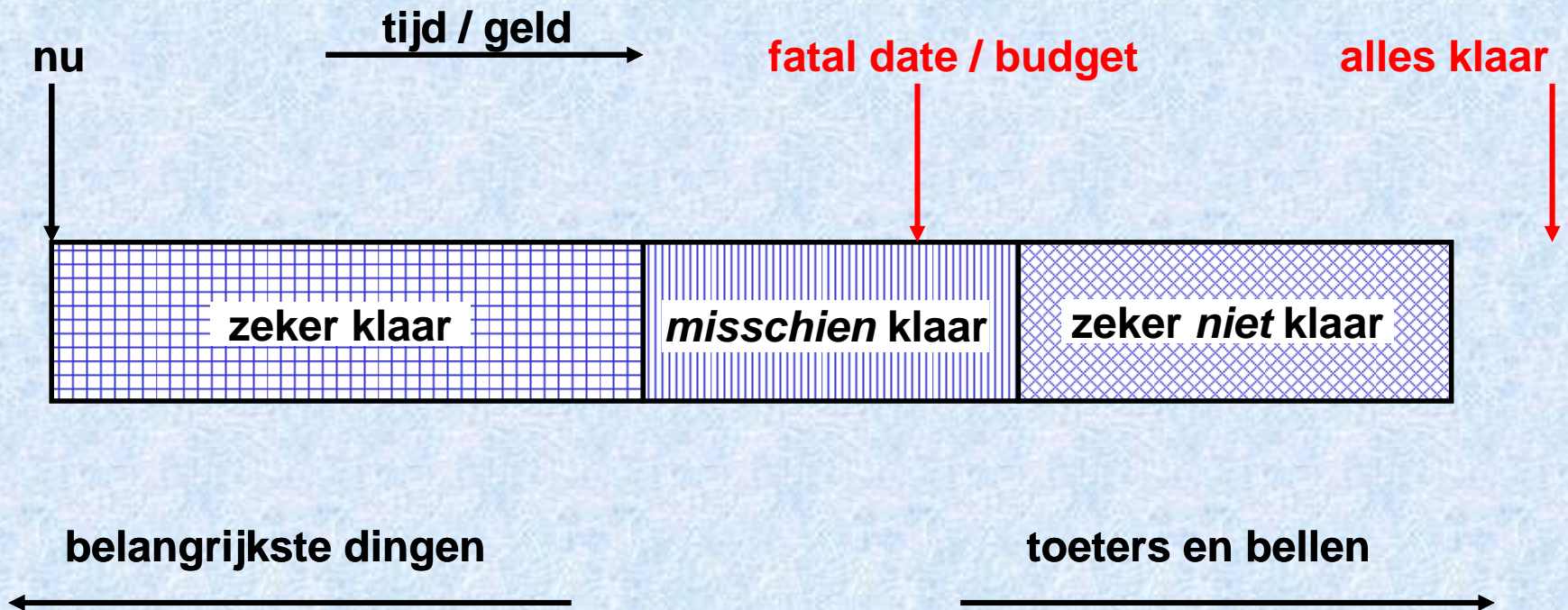


Tasks en Deliveries

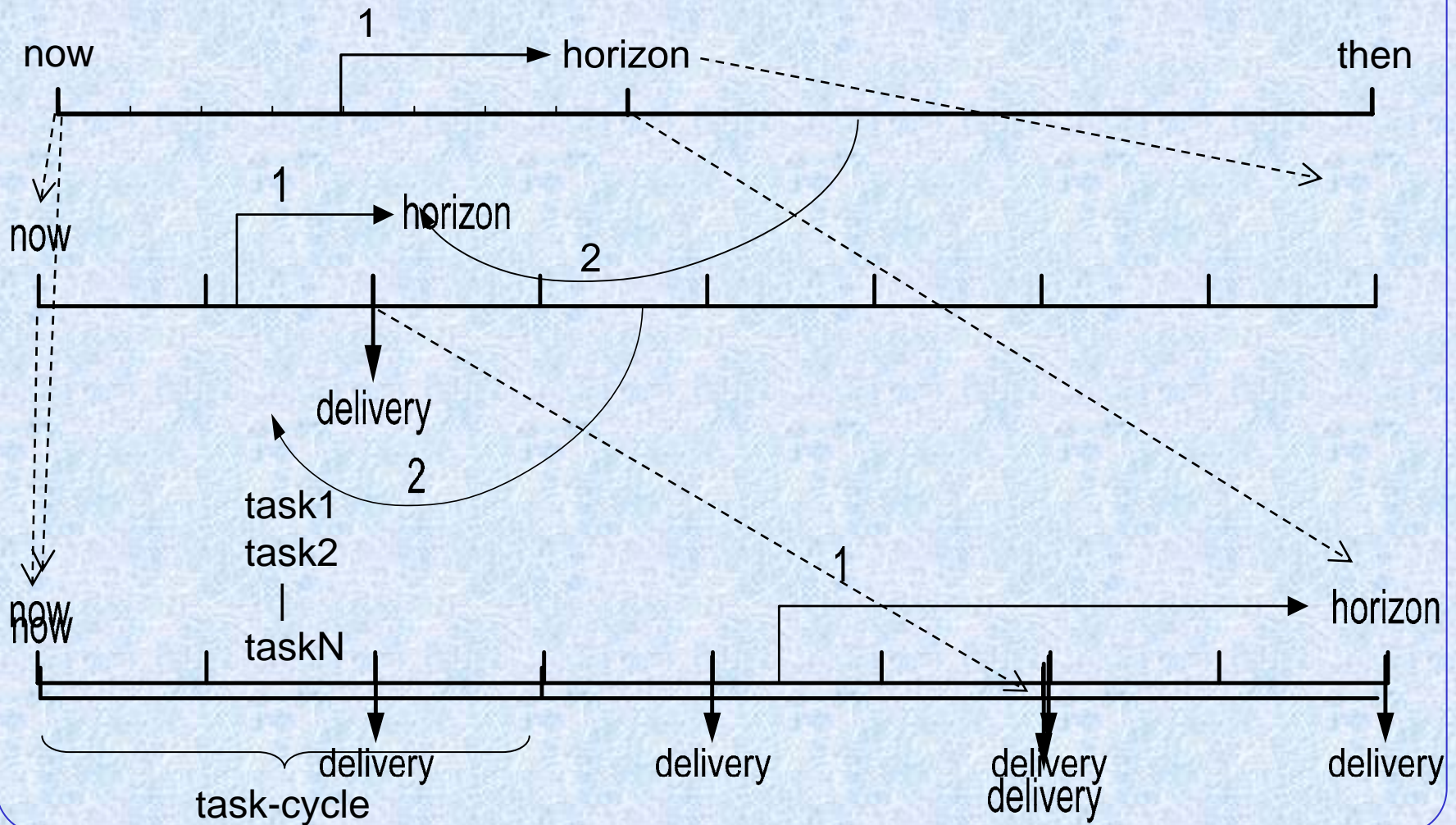


TimeLine

Wat de klant wil, kan hij zich niet veroorloven



TimeLine



Waarom Evo voor het Product ?

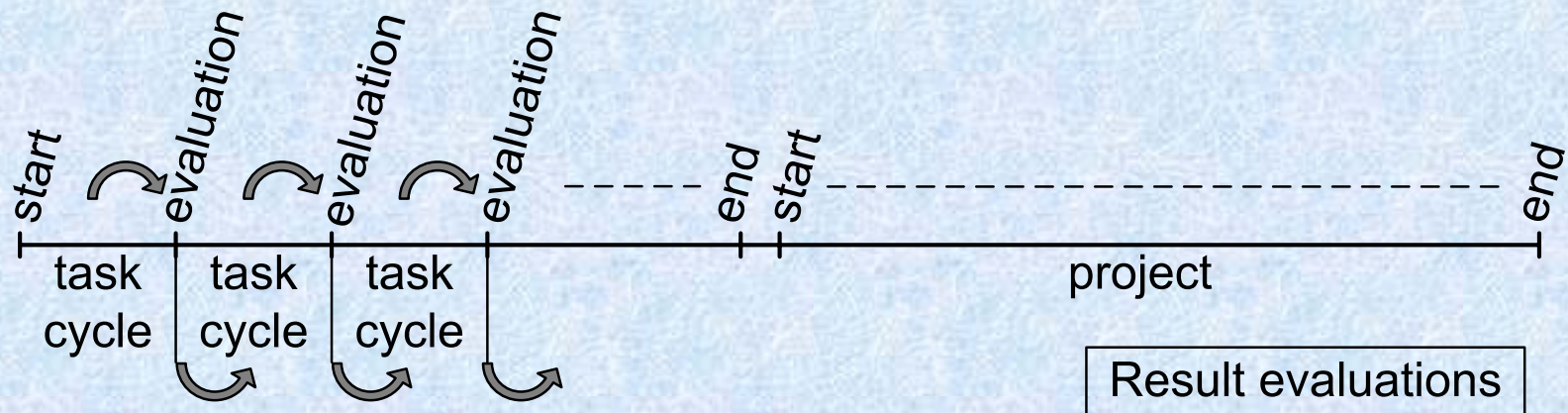
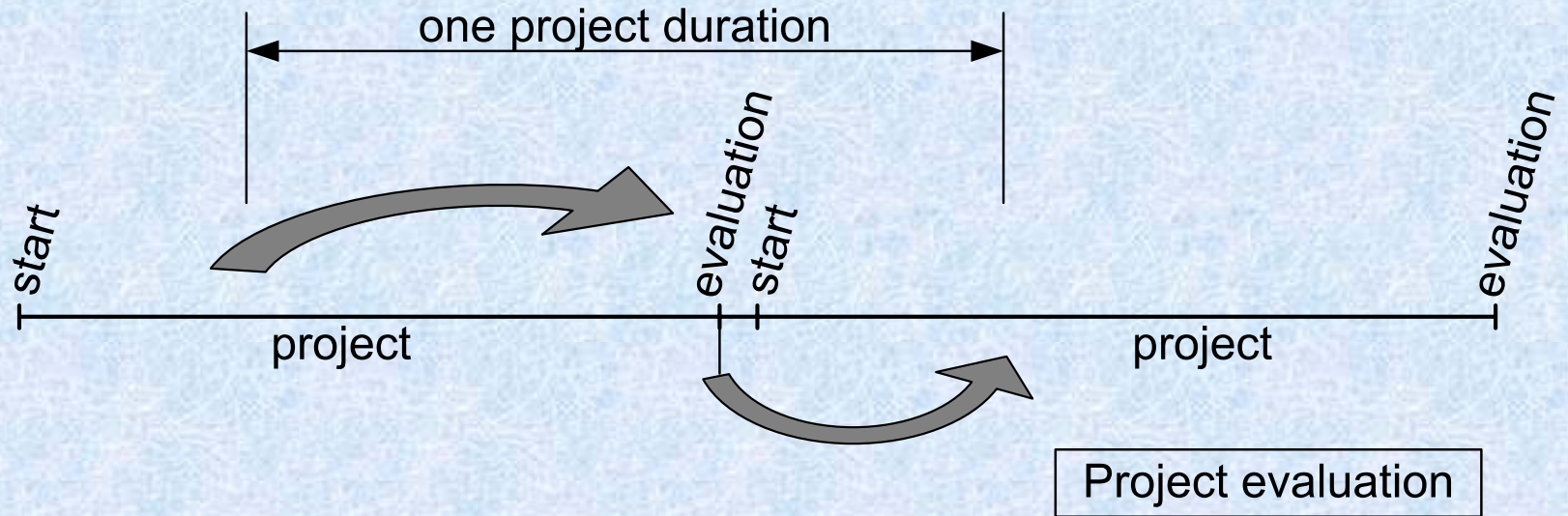
- **Wij** kennen de echte requirements niet
- **Zij** kennen de echte requirements ook niet
- **Samen** moeten we er achter komen (wees eerlijk!)
- **Wat** de klant wil, kan hij zich niet veroorloven
- **Is** wat de klant wil wel wat hij nodig heeft?
- **Mensen** neigen meer te doen dan werkelijk nodig
vooral als ze niet precies weten wat de bedoeling is

**Wanneer tijd, geld en middelen beperkt zijn,
hebben we niet het recht éénzijdig het budget overschrijden**

Waarom Evo voor het Project ?

- **Zijn we wel effectief?** (produceren we toegevoegde waarde)
- **Zijn we wel efficiënt?** (gebruiken we onze beperkte tijd optimaal)
- **Leren we actief van onze fouten?** (PDCA)
- **Hoe schatten, plannen en volgen we op?**
- **Hoe gaan we om met interrupts?**
- **Hoe gaan we om met afhankelijkheid van anderen?**
- **Leren we van project evaluaties?**

Project evaluations



Wanneer hebben we Evo *niet* nodig

- **Requirements zijn volledig duidelijk: gebruik waterval**
(= productie)
- **Requirements kunnen gemakkelijk worden gerealiseerd binnen de beschikbare tijd** (maar met Evo kan het een stuk efficiënter)
- **Iedereen weet precies wat hij moet doen**
- **De klant kan wachten tot we klaar zijn**
- **Management weet zich geen raad met de bespaarde tijd**
- **Geen Sense of Urgency**

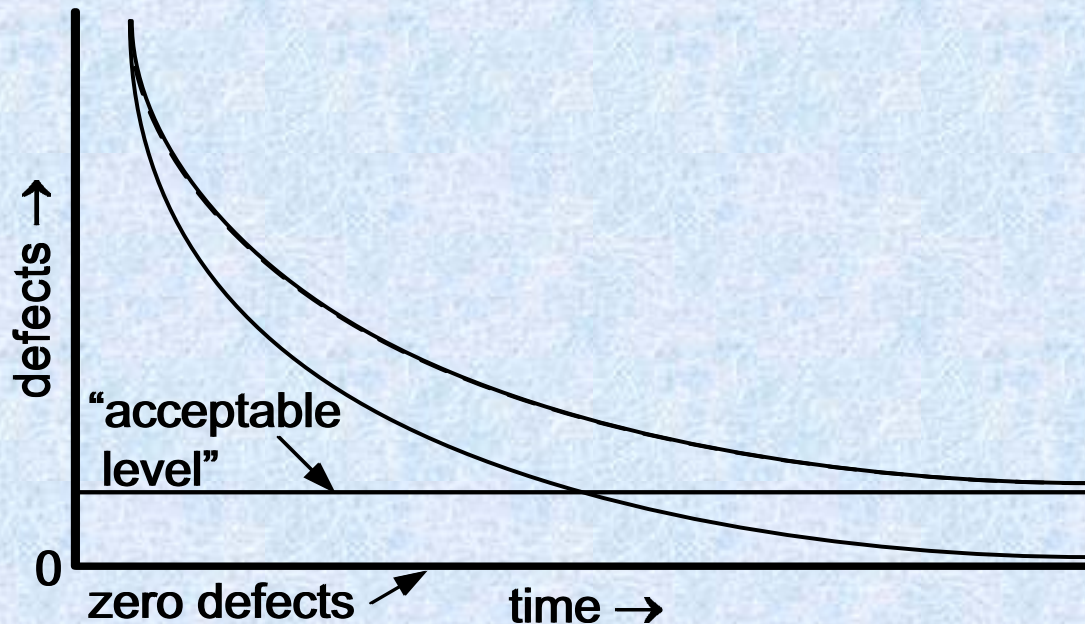
Pas Evo *alleen* toe op projecten die moeten slagen

We optimaliseren continu (PDCA)

- **Het product**
hoe we het beste product realiseren (doel !)
- **Het project**
hoe we het beste product zo effectief en efficiënt mogelijk realiseren
- **Het proces**
 - Blijven zoeken naar manieren om het beter te doen
 - Afkijken van andere methoden
 - Inpakken van methoden die beter werken
 - Opschorten van methoden die minder goed werken

Is foutloze software mogelijk?

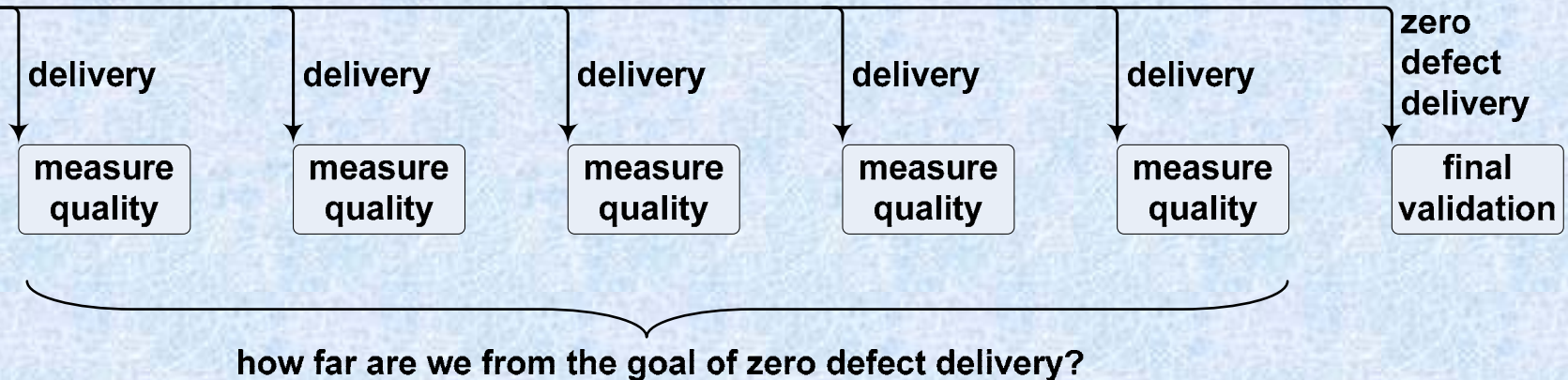
- **Zero Defects is een asymptoot**



- **Toen Philip Crosby in 1961 met Zero Defects begon, daalde het aantal gemaakte fouten vrijwel direct met 40%**

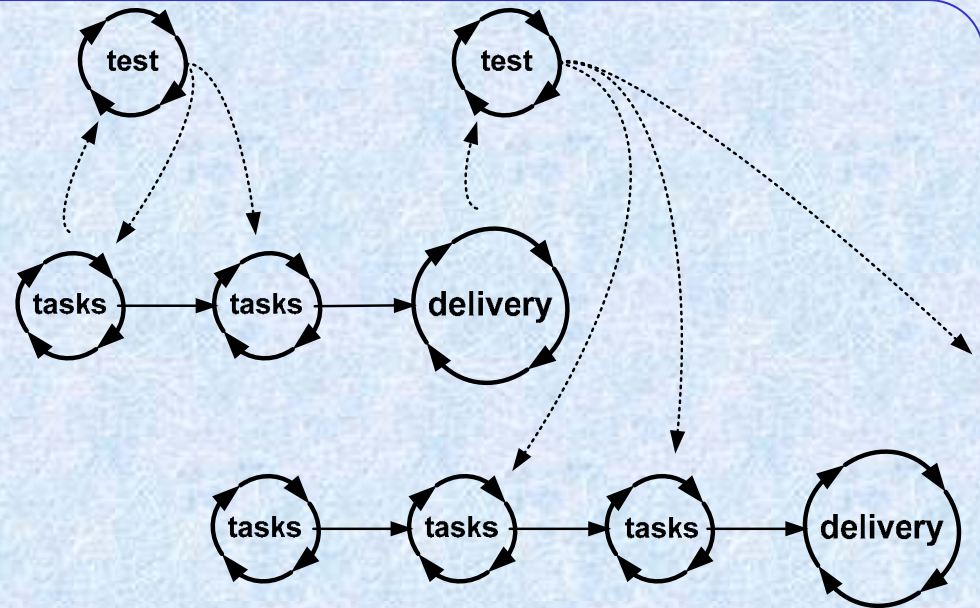
Testen in Evo

evolutionary project track



- **Uiteindelijke validatie mag geen problemen meer tegenkomen**
- **Eerdere verificaties spiegelen het kwaliteitsniveau aan de ontwikkelaars: wat moeten we nog beter leren beheersen**
- **Evo heeft *géén debugging fase!***

Evo cycli voor Testen



- Testers organiseren hun werk in wekelijkse TaskCycles
- DeliveryCycle is de Test-Feedback cyclus
- Testers hebben hun eigen TimeLine, gesynchroniseerd met de TimeLine van de ontwikkelaars
- Testers weten wat ze moeten testen
- Testers zijn vrijwel tegelijk met de ontwikkelaars klaar
- Testers checken werk van ontwikkelaars **vóór** het klaar is

Introduceren van Evo

1. Introduceren van *Tasks*

Het organiseren van het werk

→ Korte termijn

2. Introduceren van *TimeLine*

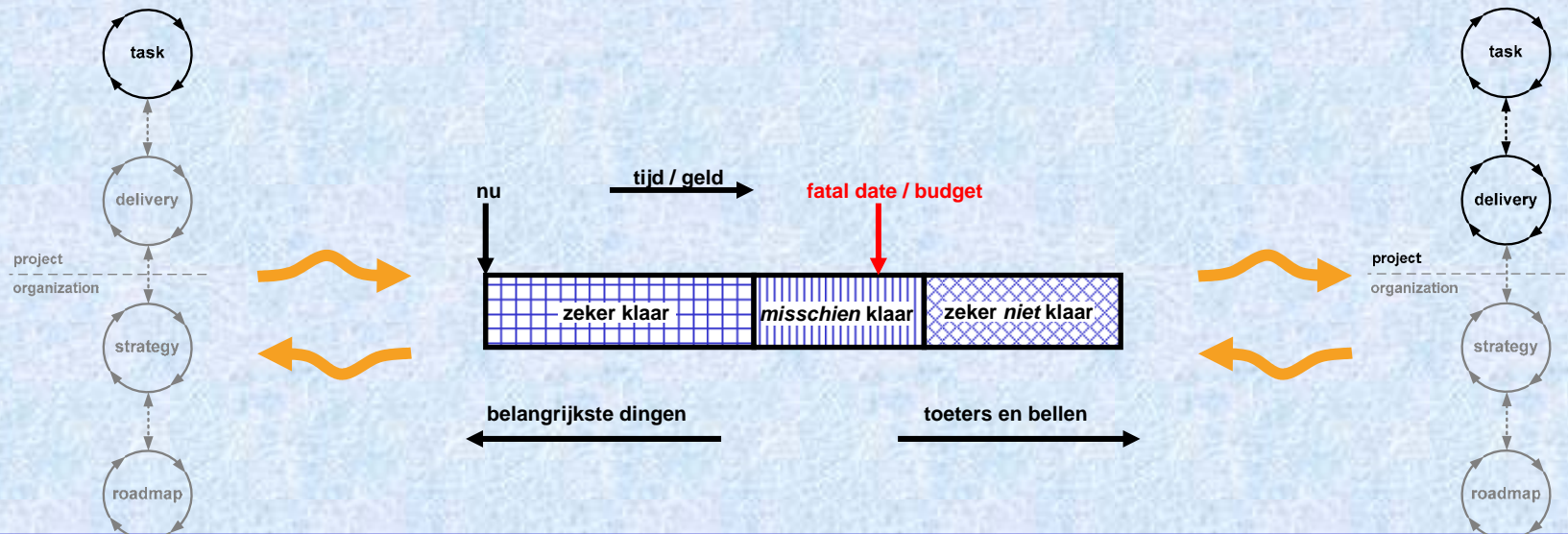
We steeds weer opnieuw
ontwerpen het project

→ Langere termijn

3. Introduceren van *Deliveries*

Focus op Resultaten en feedback

→ Verbindt lang en kort



Today

6 mei 2004 wk 19

Project

Dino-QUA

Delivery

4

☐ Other work

TaskCycle

Future

TaskType

Code

Priority

0

Who

-

hrs

hr (=Timebox!)

Plan Reviewer

-

done (Checks)

☐ 100% done

Hours of

0

total

in Cycle

0

OK

Fut

0

not OK

TaskSheet

Results

Checks

Project and Delivery

Tasks Cycle and Delivery

Timing

Printing

Edit/New

Task Name

Hoe gaan we exporteren doen

Cycle

-

☐ Other work

Task cycle due date

Delivery Nr

4

Delivery Name

Delivery 4

Delivery Due

21 mei 2004 wk 21

The TaskSheet is used to focus on what the task really is about.

Task Description

Functional Requirements

(what the result of this task should be)

Performance Requirements

(how well the result should do the what)

Constraints

(what not)

Validation

(how to check that the requirements are met)

Implementation Ideas

(solution direction ideas)

Planning

(to make sure task is done on time)

Unclears

(anything that is still unclear)

ID	Project	Delivery	Cycle	Task cycle due date	Pri	Who	hrs	Done	TaskName
59	Dino-QUA	Delivery 4	Fut		0	-			Hoe gaan we exporteren doen
58	Dino-QUA	Delivery 4	Fut		0	-			Hoe gaan we importeren doen?
212	Dino-QUA	Delivery 7	13	11 jun 2003 wk 24	5	Niko	18		Documentatie SPS, SCM-BDB
220	Dino-QUA	Delivery 6	13	11 jun 2003 wk 24	5	Ronald	6		Samples importeren
211	Dino-QUA	Delivery 7	13	11 jun 2003 wk 24	5	Niko	4		Conversie aanpassen n.a.v. Hans van der Meij
214	Dino-QUA	Delivery 6	13	11 jun 2003 wk 24	4	Arian	10		Export blokken maken
215	Dino-QUA	Delivery 6	13	11 jun 2003 wk 24	5	Arian	2		Checkbox toevoegen voor export-blokken
216	Dino-QUA	Delivery 6	13	11 jun 2003 wk 24	5	Arian	2		Backsupport toevoegen met Ronald
217	Dino-QUA	Delivery 6	13	11 jun 2003 wk 24	5	Ronald	2		Backsupport toevoegen met Arian
218	Dino-QUA	Delivery 6	13	11 jun 2003 wk 24	5	Arian	6		Uitzoeken rechts invullen van kolommen bij sample, subsample
219	Dino-QUA	Delivery 6	13	11 jun 2003 wk 24	5	Ronald	6		Maken Process dialog
210	Dino-QUA	Delivery 7	13	11 jun 2003 wk 24	5	Niko	2		Conversie aanpassen voor Omrekenfactor koppeling
200	Dino-QUA	Delivery 4	12	4 jun 2003 wk 23	5	Niko	4	OK	parameterformulier voor analyserapport met tabbladen
201	Dino-QUA	Delivery 4	12	4 jun 2003 wk 23	5	Arian	3	OK	Aanpassingen Monsterscherm doorvoeren (nieuwe velden)
104	Dino-QUA	Delivery 5	12	4 jun 2003 wk 23	5	Niko	4	OK	Uitbreiden datamodel met QUA, CALIBITY en aanpassing van de...

Mijn project is anders

- In *elk* project is er aanvankelijk altijd iemand die zegt:
“Mooi verhaal, maar *mijn* project is anders.
Het is onmogelijk om elke twee weken iets op te leveren”
- In *elk* project, kost het dezelfde persoon minder dan een uur om de eerste korte Deliveries te definiëren
- Dit is aanvankelijk een van de moeilijker zaken van Evo
We moeten als het ware een schakelaar leren omzetten

Case: Amerikaans bedrijf

- **Begonnen met 15 personen uit een project van 40** (rustig aan beginnen)
- **We *ontwierpen* de Evolutionaire Introductie van Evo**
- **Na enkele weken ging de rest van het project meedoen**
- **Inclusief een team in India**
- **Milestones werden vervolgens steeds gehaald**
(gemiddelde overschrijding vóór Evo was 20%)
- **Prijs voor het beste team**
- **Product manager: big smile:** “This product had the fastest time-to-market with the highest quality at introduction of any platform in our group in more than 10 years.”
- **Momenteel werkt de gehele ontwikkelafdeling (>120) met Evo**

Evo werkt ook in grotere en multi-site projecten

Links

- <http://www.gilb.com>
Tom Gilb's website: Evo guru
- <http://www.malotaux.nl/nrm/English>
Niels Malotaux' website: Evo evangelist
- <http://www.malotaux.nl/nrm/Evo>
Evo pages
- <http://www.malotaux.nl/nrm/pdf/MxEvo.pdf>
Booklet: Evolutionary Project Management Methods (issues and 2001 experience)
- <http://www.malotaux.nl/nrm/pdf/Booklet2.pdf>
Booklet: How Quality is Assured by Evolutionary Methods
(more recent very practical implementation experience)
- <http://www.malotaux.nl/nrm/pdf/EvoTesting.pdf>
Booklet: Optimizing the Contribution of Testing to Project Success (how to integrate testing in Evo)
- <http://www.malotaux.nl/nrm/pdf/EvoQA.pdf>
Booklet: Optimizing Quality Assurance for Better Results
(same as testing booklet, but now for non-software environments)
- <http://www.malotaux.nl/nrm/pdf/EvoRisk.pdf>
Booklet: Controlling Project Risk by Design
- <http://www.malotaux.nl/nrm/Evo/ETAF.htm>
Download the Evo Task Administrator (ETA) tool (expects MSAccess2000~2003)

Wie gaat vanaf morgen anders werken?

Niels Malotaux

N R Malotaux
Consultancy

+31-30-228 88 68

niels@malotaux.nl

www.malotaux.nl/nrm/English

Case: Nederlandse organisatie

- **‘AGILE’ ontwikkeling (XP) - 2000**
 - Over budget en tijd door informele en directe communicatie met stakeholders (wensenlijst te lang en niet gemanaged)
- **Waterval gebaseerd - 2000/2003**
 - Beginnen met uitgebreide, “complete” documentatie
 - Ontwikkelen op basis van één vast SRS/Design document
 - Over budget en tijd door problemen met planning en requirements wijzigingen bij aflevering
- **Evo - 2003**
 - Module gereed in een verbazend korte tijd
 - Stakeholder betrokkenheid veel beter (beter gemanaged)
 - Eind resultaat veel meer volgens de verwachting
 - Software kwaliteit niet minder, ondanks veel minder testinspanning

SW groepsleider

- Ik heb vele project management cursussen gevolgd
- Deze gingen alle voorbij aan menselijke gedrags-elementen, focus, continue toepassen van de PDCA cyclus, het leveren van stakeholder value in korte cycli, welbewust uitlokken van feedback
- Alle problemen waar Evo oplossingen voor geeft worden in boeken wel genoemd. Oplossingen worden echter gezocht in allerlei formele controle-mechanismen, die de kern van de problemen niet oplossen